

ESP Normalization Exercise Solution

Customer Details View

Initial Table

CustomerNumber, Name, Address, City, Province, PostaleCode, HomePhone

1NF

CustomerNumber, FirstName, LastName, Address, City, Province, PostaleCode, HomePhone

2NF

CustomerNumber, FirstName, LastName, Address, City, Province, PostaleCode, HomePhone

3NF

CustomerNumber, FirstName, LastName, Address, City, Province, PostaleCode, HomePhone

View ERD

Customer

CustomerNumber
FirstName
LastName
Address
City
Province
PostalCode
Phone

Customer Orders View

Initial Table

CustomerNumber, CustomerName, Address, Phone, Date, **OrderNumber**,
(ItemNumber, Description, Quantity, SellPrice, CurrentPrice, Amount), Subtotal ,
GST, Total

1NF

CustomerNumber, CustomerFirstName, CustomerLastName, Address, City, Province,
PostalCode, Phone, Date, **OrderNumber**, Subtotal , GST, Total

OrderNumber, **ItemNumber**, Description, Quantity, SellPrice, CurrentPrice,
Amount

2NF

CustomerNumber, CustomerFirstName, CustomerLastName, Address, City, Province,
PostalCode, Phone, Date, **OrderNumber**, Subtotal , GST, Total

OrderNumber, **ItemNumber**, Quantity, SellPrice, Amount

ItemNumber, Description, CurrentPrice

3NF

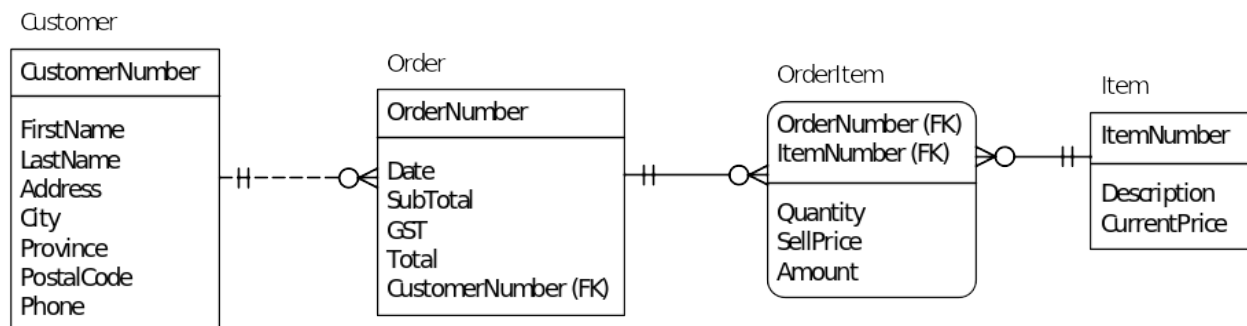
CustomerNumber, Date, **OrderNumber**, Subtotal , GST, Total

CustomerNumber, CustomerFirstName, CustomerLastName, Address, City,
Province, PostalCode, Phone

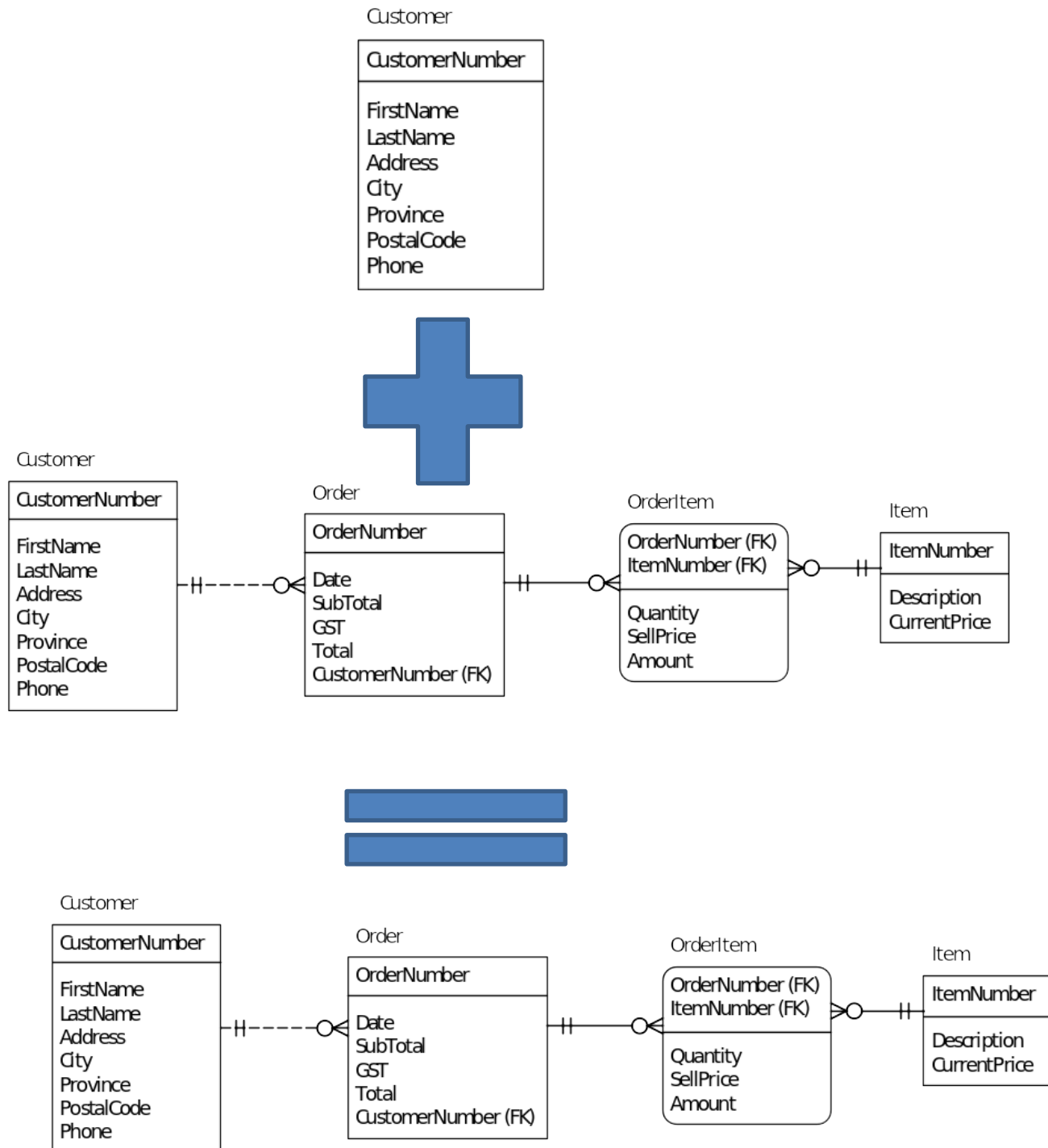
OrderNumber, **ItemNumber**, Quantity, SellPrice, Amount

ItemNumber, Description, CurrentPrice

View ERD



Merge Customer Orders View with current ERD



In the Customer Orders View we discovered three new entities and a number of attributes that relate to what we discovered from the Customer Details view. If any new attributes about a customer were discovered they would be added to the Customer entity.

Customer Payments Against Orders

Initial Table

OrderNumber, OrderDate, OrderTotal, CustomerName, CustomerNumber,
(PaymentDate, PaymentAmount, PaymentNumber, BalanceOwing, PaymentType,
DepositBatchNumber)

1NF

OrderNumber, OrderDate, OrderTotal, CustomerName, CustomerNumber

OrderNumber, PaymentDate, PaymentAmount, **PaymentNumber**, BalanceOwing,
PaymentType, DepositBatchNumber

2NF

OrderNumber, OrderDate, OrderTotal, CustomerName, CustomerNumber

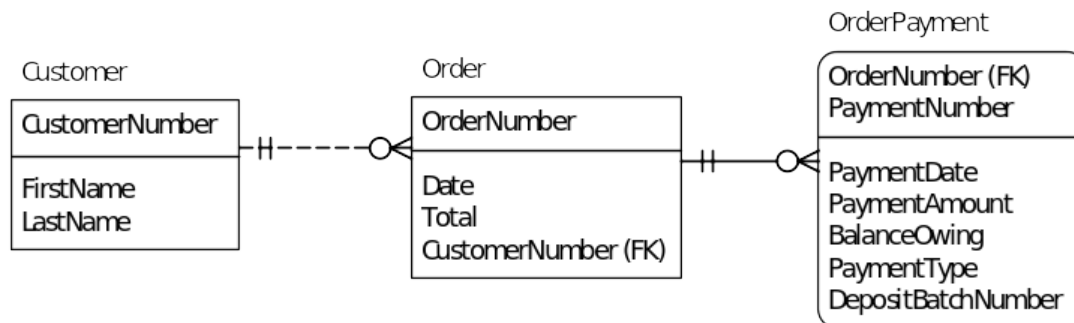
OrderNumber, PaymentDate, PaymentAmount, **PaymentNumber**, BalanceOwing,
PaymentType, DepositBatchNumber

3NF

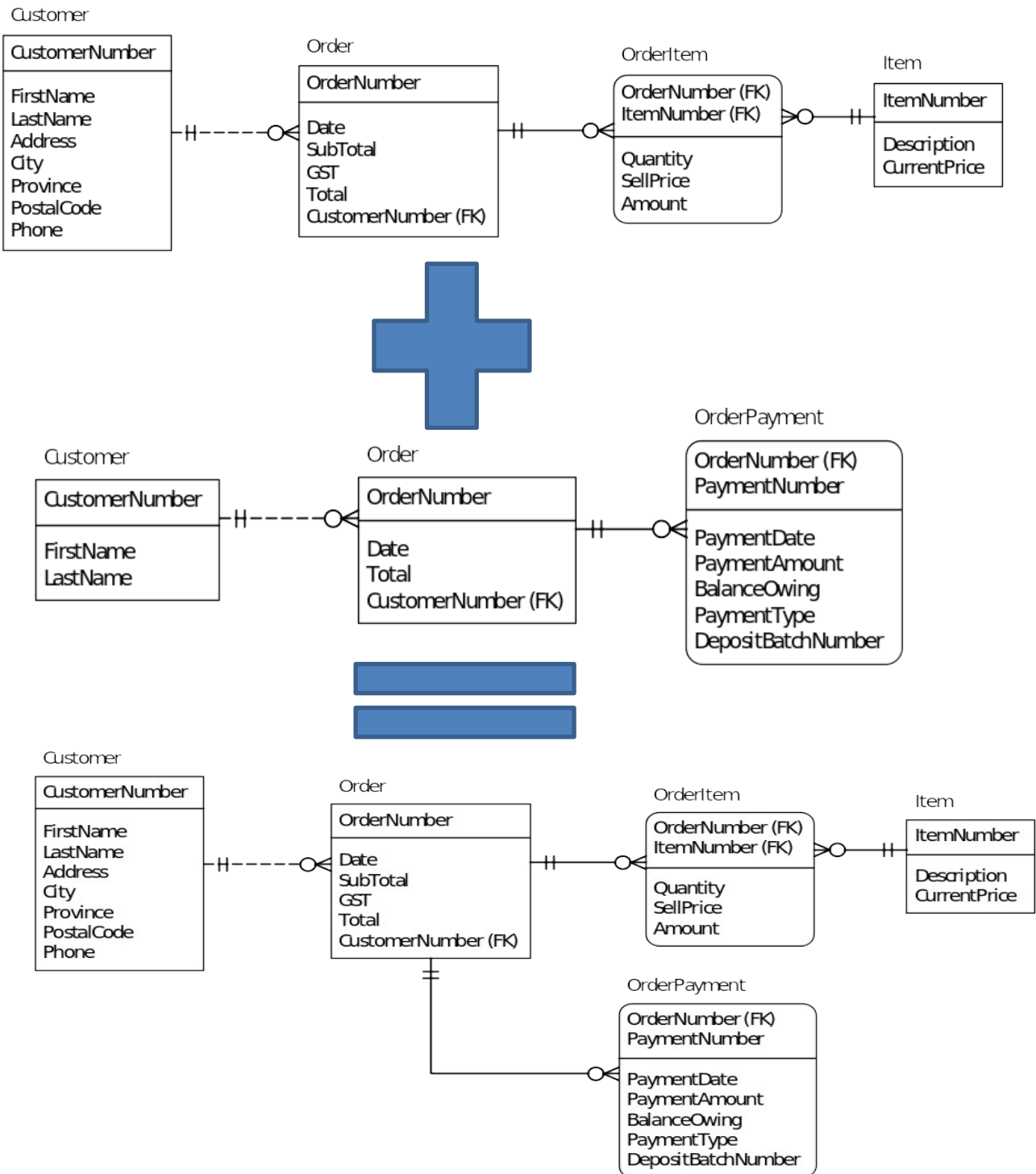
OrderNumber, OrderDate, OrderTotal, *CustomerNumber*

CustomerNumber, CustomerFirstName, CustomerLastName

OrderNumber, PaymentDate, PaymentAmount, **PaymentNumber**, BalanceOwing,
PaymentType, DepositBatchNumber



Merge Customer Payments Against Orders View with current ERD



In the Customer Payments Against Orders View we discovered a new OrderPayment entity. No new attributes were discovered for the existing entities.

Inventory Control

Initial Table

ItemNumber, ItemDescription, CurrentSalePrice, QOH, ReorderLvl (OrderDate, SupplierNumber, PONumber, Quantity, Cost)

1NF

ItemNumber, ItemDescription, CurrentSalePrice, QOH, ReorderLvl

OrderDate, SupplierNumber, **PONumber**, Quantity, Cost, *ItemNumber*

2NF

ItemNumber, ItemDescription, CurrentSalePrice, QOH, ReorderLvl

PONumber, Quantity, Cost, *ItemNumber*

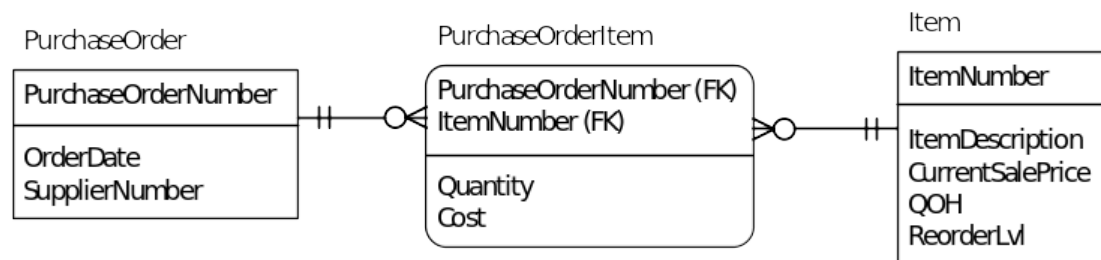
PONumber, OrderDate, SupplierNumber

3NF

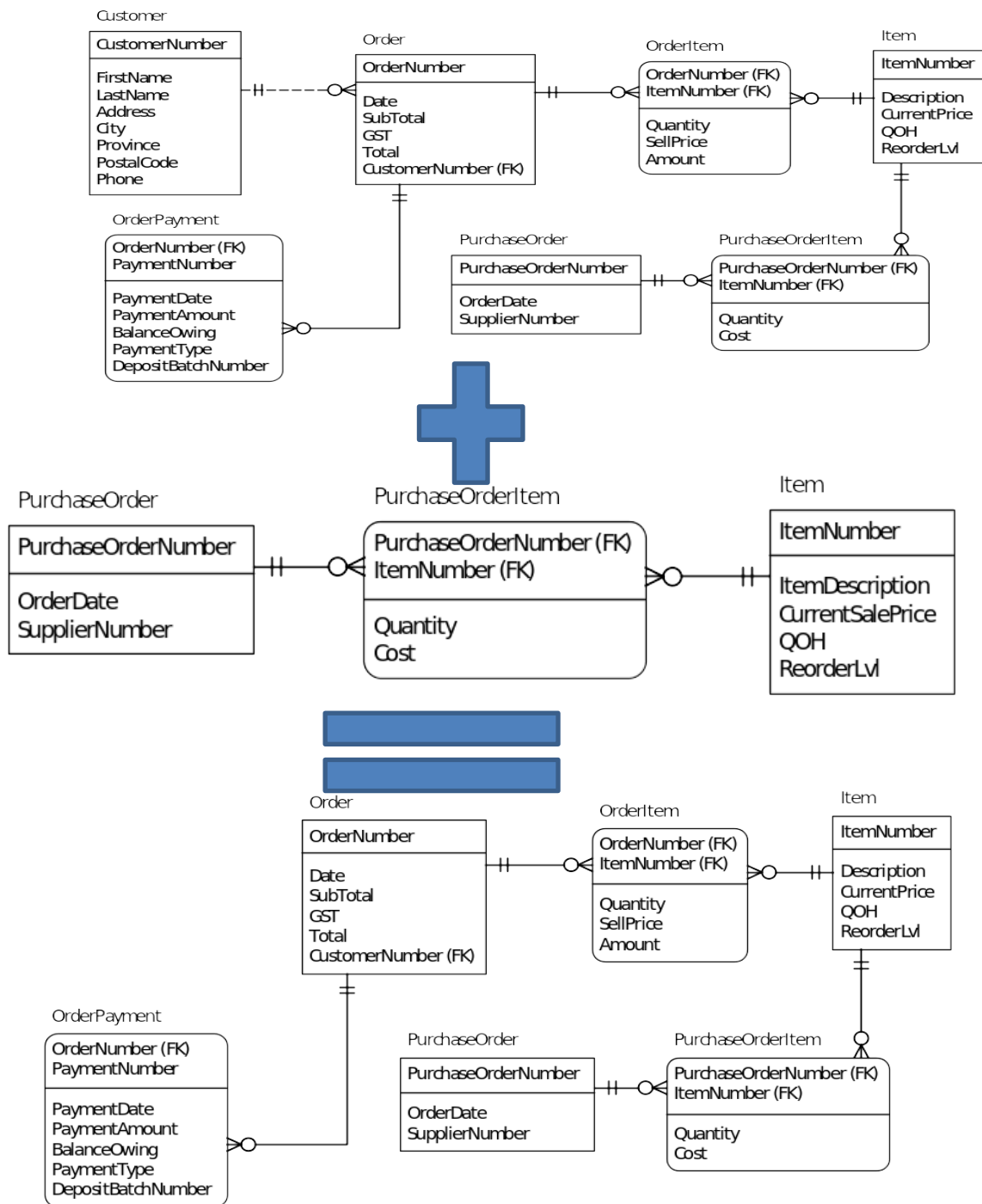
ItemNumber, ItemDescription, CurrentSalePrice, QOH, ReorderLvl

PONumber, Quantity, Cost, *ItemNumber*

PONumber, OrderDate, SupplierNumber



Merge Inventory Control View with current ERD



In the Inventory Control View we discovered 2 new entities (PurchaseOrder and PurchaseOrderItem). We also see that they merge to the existing design through the Item entity. No new attributes were discovered for the existing entities, however, the different views contained the same attribute (CurrentPrice and CurrentSalePrice) under different names. This can happen quite often since the data sometimes has a different context in different views and therefore may be called something different. You would just need to choose one name for the

Purchase Orders

Initial Table

PurchaseOrderNumber, SupplierNumber, SupplierName, Address, City, Phone, Date, (Item#, SuppliersItemNumber, SuppliersDescription, Qty, Cost, Amount), SubTotal, GST, Total

1NF

PurchaseOrderNumber, SupplierNumber, SupplierName, Address, City, Province, PostalCode, Phone, Date, SubTotal, GST, Total

PurchaseOrderNumber, Item#, SuppliersItemNumber, SuppliersDescription, Qty, Cost, Amount

2NF

PurchaseOrderNumber, SupplierNumber, SupplierName, Address, City, Province, PostalCode, Phone, Date, SubTotal, GST, Total

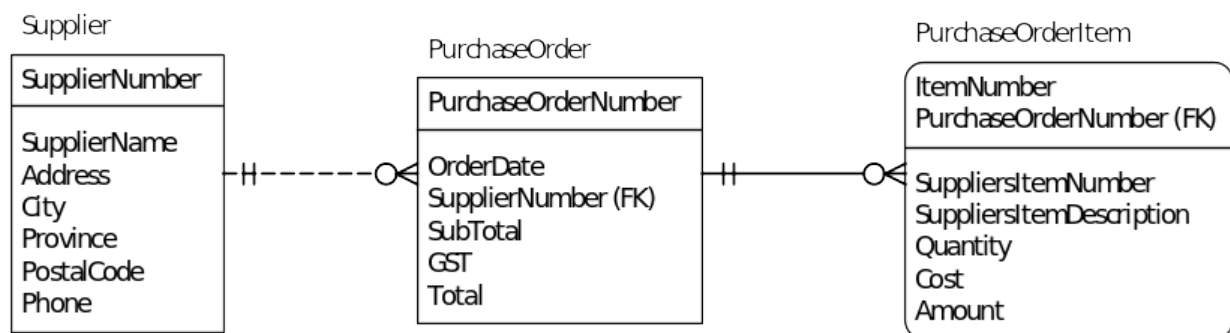
PurchaseOrderNumber, Item#, SuppliersItemNumber, SuppliersDescription, Qty, Cost, Amount

3NF

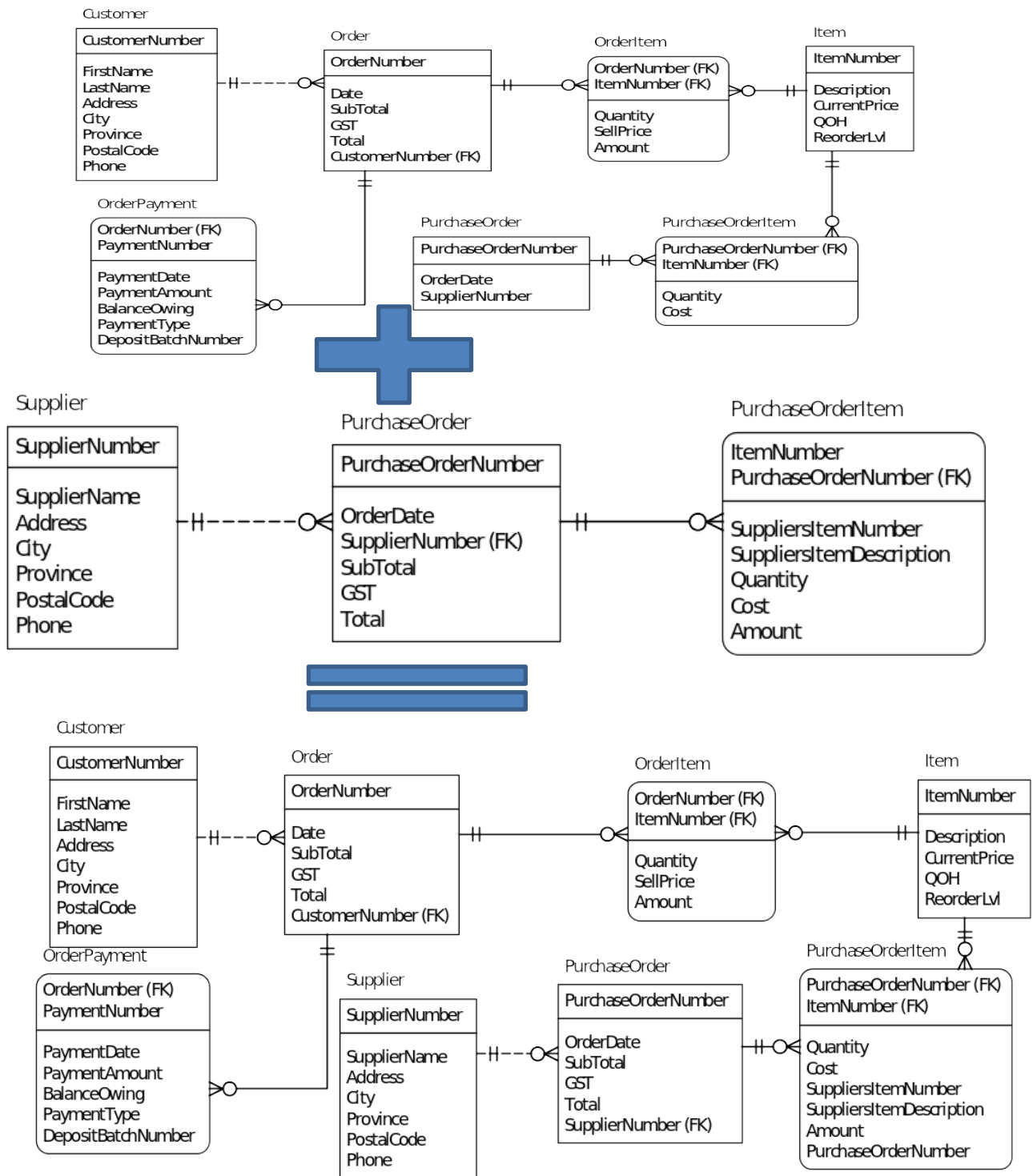
PurchaseOrderNumber, SupplierNumber, Date, SubTotal, GST, Total

SupplierNumber, SupplierName, Address, City, Province, PostalCode, Phone,

PurchaseOrderNumber, Item#, SuppliersItemNumber, SuppliersDescription, Qty, Cost, Amount



Merge Purchase Order View with current ERD



In the Purchase Order View we discovered a new Supplier entity that merge to the existing design through the Purchase Order Entity. As well, a number of new attributes for PurchaseOrder and PurchaseOrderItem were discovered.

Purchase Order Payments

Initial Table

Date, **ChequeNumber**, ChequeAmount, SupplierNumber, SupplierName,
(PurchaseOrderNumber, PurchaseOrderDate)

1NF

Date, **ChequeNumber**, ChequeAmount, SupplierNumber, SupplierName

ChequeNumber, **PurchaseOrderNumber**, PurchaseOrderDate

2NF

Date, **ChequeNumber**, ChequeAmount, SupplierNumber, SupplierName

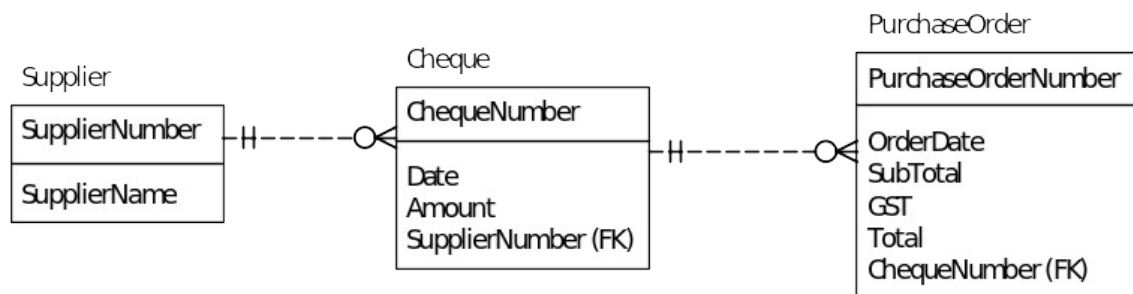
ChequeNumber, **PurchaseOrderNumber**, PurchaseOrderDate

3NF

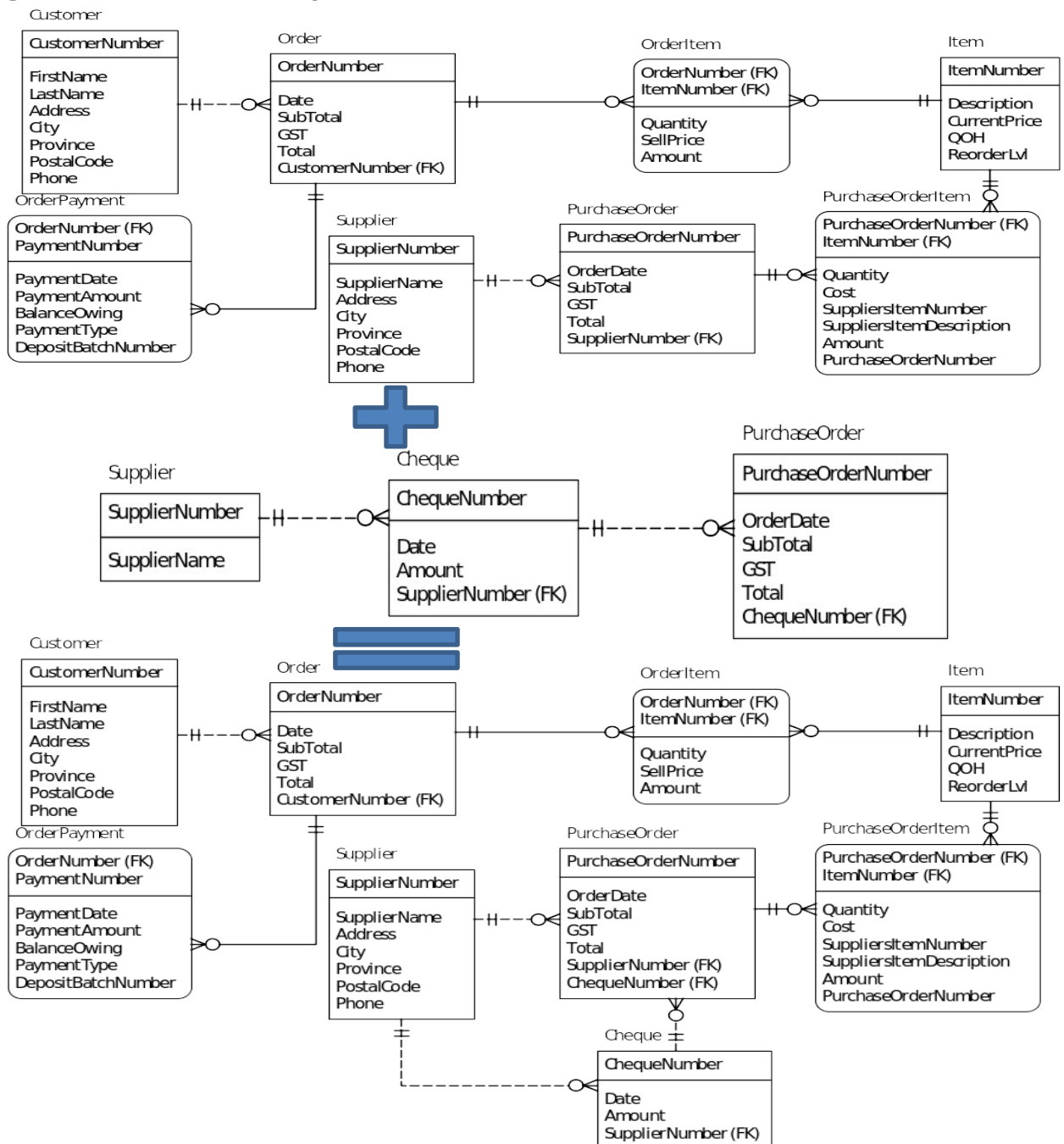
Date, **ChequeNumber**, ChequeAmount, *SupplierNumber*

SupplierNumber, SupplierName

ChequeNumber, **PurchaseOrderNumber**, PurchaseOrderDate



Merge Purchase Order Payments View with current ERD



In the Purchase Order Payments View we discovered a new Cheque entity that merges to the existing design through the Supplier Entity AND the Purchase Order.

This solution introduces a “circle” of relationships between Supplier, PurchaseOrder and Cheque. Depending on the businesses needs one of the relationships (and its corresponding foreign key) might be able to be removed. Could this be possible here?

This is the final database design that will support the operations and business